CoMelSinger: Discrete Token-Based Zero-Shot Singing Synthesis With Structured Melody Control and Guidance

Junchuan Zhao, Wei Zeng, Tianle Lyu, Ye Wang, *Member, IEEE*

***Abstract*—Singing Voice Synthesis (SVS) aims to generate expressive vocal performances from structured musical inputs such as lyrics and pitch sequences. While recent progress in discrete codec-based speech synthesis has enabled zero-shot generation via in-context learning, directly extending these techniques to SVS remains non-trivial due to the requirement for precise melody control. In particular, prompt-based generation often introduces prosody leakage, where pitch information is inadvertently entangled within the timbre prompt, compromising controllability. We present CoMelSinger, a zero-shot SVS framework that enables structured and disentangled melody control within a discrete codec modeling paradigm. Built on the non-autoregressive MaskGCT architecture, CoMelSinger replaces conventional text inputs with lyric and pitch tokens, preserving in-context generalization while enhancing melody conditioning. To suppress prosody leakage, we propose a coarse-to-fine contrastive learning strategy that explicitly regularizes pitch redundancy between the acoustic prompt and melody input. Furthermore, we incorporate a lightweight encoder-only Singing Voice Transcription (SVT) module to align acoustic tokens with pitch and duration, offering fine-grained frame-level supervision. Experimental results demonstrate that CoMelSinger achieves notable improvements in pitch accuracy, timbre consistency, and zero-shot transferability over competitive baselines. Audio samples are available at https://danny-nus.github.io/CoMelSinger/.**

***Index Terms*—Singing voice synthesis, zero-shot singing voice synthesis, voice cloning, neural codecs, deep learning, masked generative models.**

I. Introduction

SINGING voice synthesis (SVS) aims to transform structured musical inputs—most often lyrics and pitch sequences—into expressive, high-quality vocal performances. Over the past decade, it has moved from a niche research topic to an essential tool in creative audio technologies, propelled by the rise of AI-driven music generation, virtual performers, and personalized media experiences. Its applications now extend well beyond traditional karaoke systems, finding a place in virtual idol production, game soundtracks, and content creation for social platforms. Parallel to these expanding use cases, advances in deep generative models have brought marked gains in timbre fidelity, pitch accuracy, and the overall naturalness of synthesized voices [1], [2], [3], [4], [5], [6], [7], [8].

Recent SVS frameworks, including end-to-end [6], [9] and diffusion-based architectures [3], [4], [8], [10], [11], have demonstrated strong performance in supervised scenarios with seen singers. Nevertheless, zero-shot SVS [12] [13], [10], [14]—synthesizing singing voices for unseen speakers without additional fine-tuning—still has substantial room for improvement.

Junchuan Zhao, Wei Zeng, Tianle Lyu, Ye Wang, are affiliated with the School of Computing, National University of Singapore, Singapore. Ye Wang is the correspondence author of this paper. Contact junchuan@comp.nus.edu.sg for further questions about this work.

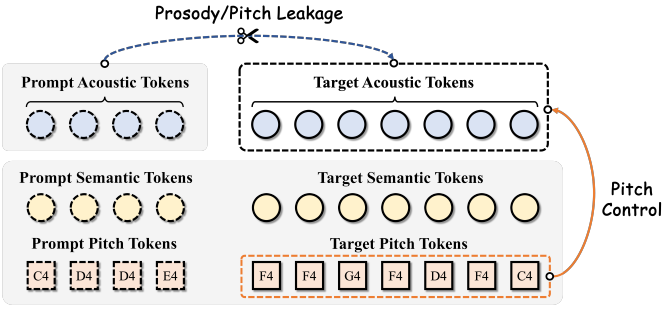


Fig. 1: Illustration of pitch leakage in prompt-based SVS. Despite conditioning on lyrics (semantic tokens) and pitch tokens, the model may still infer prosodic cues from prompt acoustic tokens, leading to pitch or prosody leakage.

Discrete token-based architectures show great in-context learning capabilities and provide a promising pathway for such zero-shot generation. In a related task, text-to-speech (TTS) has witnessed rapid progress with discrete acoustic tokens derived from vector quantization and neural audio codecs [15], [16], [17], [18], [19]. By mapping complex waveforms into a quantized latent space, these tokens capture timbre, prosody, and phonetic content, thereby reformulating speech synthesis as symbolic sequence modeling akin to language modeling. The success of token-based TTS systems is largely enabled by large-scale multi-speaker corpora, which provide sufficient diversity to learn robust and generalizable representations. In contrast, the scarcity and limited diversity of singing data make token-based modeling for SVS significantly more challenging.

Built upon this data-rich foundation, recent TTS systems [15], [18], [20], [21], [16], [17], have adopted large language model (LLM)-style architectures to model the conditional distribution of acoustic tokens given phoneme sequences and optional prompts. Within this framework, in-context learning (ICL) becomes feasible: a short segment of reference speech, represented as discrete tokens, serves as an acoustic prompt to guide synthesis in terms of speaker identity and style. This approach, exemplified by models such as VALL-E [15], enables zero-shot speech synthesis by treating speech generation as a form of conditional codec language modeling, without requiring speaker labels or model adaptation. Inspired by these advances, researchers have begun exploring discrete token-based methods for SVS.

However, directly extending TTS by replacing textual input with structured musical inputs—such as lyrics and pitch tokens—while reusing the same modeling pipeline reveals a unique challenge in the singing domain: prosody leakage from the acoustic prompt, as illustrated in Figure 1. In prompt-based synthesis, the acoustic prompt is intended to provide timbral cues, yet pitch-related attributes—including contour and timing—are often inadvertently encoded into its latent representation. This unintended encoding leads to timbre–melody entanglement, where the prompt simultaneously influences vocal timbre and melodic realization. As a result, the system's control over the explicitly specified pitch sequence is weakened, undermining the precise separation between timbre conditioning and melody generation that SVS requires.

The disparity in dataset size between singing [22], [23], [24], and speech [25], [26], [27], further exacerbates the difficulty of addressing prosody leakage. Importantly, the effectiveness of in-context learning in TTS relies on large-scale, diverse speech corpora, which enable robust learning of disentangled token representations. In contrast, singing datasets are generally smaller and less varied, making it harder to avoid prosodic interference and to generalize prompt-based conditioning. Consequently, directly transferring token-based prompting strategies from TTS to SVS often leads to weaker melody control in zero-shot scenarios. In this case, achieving reliable melody control is particularly crucial for maintaining alignment with the musical score.

Moreover, SVS requires much finer control over pitch and melody than TTS or voice conversion, as the generated singing must accurately follow the musical score while preserving timbre. Although attribute control has been explored through adversarial training [19], contrastive learning [28] [29], and information-bottleneck methods [30] [31] [32] [33], these approaches primarily focus on coarse prosodic patterns or emotional cues and provide limited support for fine-grained melody control. Make-A-Voice [12], as a representative discrete-token SVS system, adopts prompt-guided conditioning but lacks explicit mechanisms to prevent the acoustic prompt from influencing melody realization. Recent unified speech-and-singing models such as Vevo 1.5[1]/2.0 [34] take a representation-level perspective by introducing melody- or prosody-aware tokenizers that explicitly model musical F0 structures, aiming to reduce melody leakage through improved disentanglement. However, these approaches typically rely on external melody audio (e.g., piano or humming recordings) as conditioning signals, which can introduce alignment ambiguities between melody and lyrics and limit the precision of fine-grained melody control.

To address the challenges of prosody leakage and limited melody controllability in zero-shot SVS, we propose CoMelSinger, a discrete codec-based framework with structured melody control. CoMelSinger builds on the non-autoregressive MaskGCT architecture [18], adapting it to accept musical inputs consisting of lyrics and pitch tokens. To achieve better melody–timbre control, we introduce a coarse-to-fine contrastive learning strategy that limits excessive pitch-related information in the acoustic prompt, allowing the explicit pitch condition to guide melodic realization more effectively. We further incorporate a lightweight, encoder-only Singing Voice Transcription (SVT) module to provide fine-grained, frame-level supervision by aligning acoustic tokens with pitch and duration sequences. Together, these designs enable accurate melody modeling, maintain timbre consistency, and preserve the in-context learning capability of discrete token-based systems. Extensive experiments on both seen and unseen singers demonstrate that CoMelSinger delivers substantial improvements in pitch accuracy, timbre consistency, and overall synthesis quality compared with state-of-the-art SVS baselines in zero-shot scenarios. The main contributions of this work are:

- We propose CoMelSinger, a discrete token-based SVS framework for zero-shot synthesis with structured melody control.
- We introduce a coarse-to-fine contrastive learning mechanism to improve melody–timbre control by limiting excessive pitch-related information in the acoustic prompt.
- We develop a lightweight SVT module that aligns acoustic tokens with pitch and duration, providing frame-level supervision to improve melody fidelity.
- Comprehensive experiments on public SVS datasets demonstrate that CoMelSinger achieves superior pitch accuracy, timbre consistency, and generalization to unseen singers.

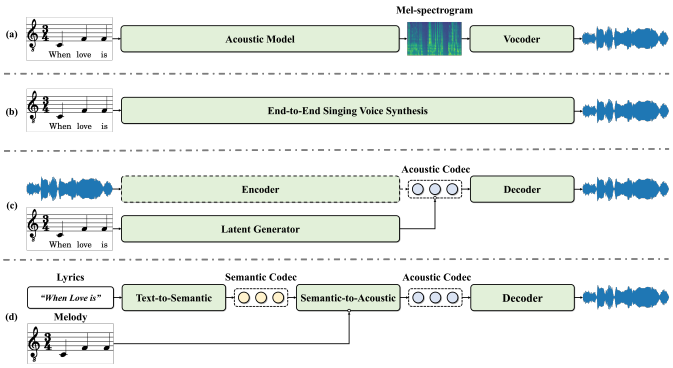


Fig. 2: Comparison of SVS system architectures. (a) Two-stage pipeline using a pre-defined continuous intermediate representation. (b) End-to-end system mapping directly from musical input to audio. (c) HiddenSinger-style [3] system employing a pretrained audio codec, with frozen components during codec prediction and audio synthesis (dashed outlines). (d) Three-stage SVS systems such as Make-a-Voice [12] and our proposed CoMelSinger, utilizing discrete intermediate representations.

II. Related Works

A. Singing Voice Synthesis

Singing voice synthesis (SVS) aims to produce expressive vocal performances from structured musical inputs such as note pitch, duration, and lyrics. Compared with text-to-speech (TTS), SVS presents unique challenges, including a wider

[1]https://github.com/open-mmlab/Amphion/tree/main/models/svc/vevosing

pitch range and sustained phonation, which demand fine-grained melody modeling. Figure 2 illustrates the evolution of representative SVS architectures, from continuous-feature pipelines to end-to-end models and, more recently, discrete codec-based frameworks. Early SVS systems relied on unit-selection synthesis (e.g., VOCALOID [35], [36]) or statistical approaches based on hidden Markov models (HMMs) [37]. With the advent of deep learning, SVS models increasingly adopted a two-stage architecture—comprising an acoustic model followed by a vocoder—including XiaoiceSing [38], DeepSinger [39], and Sinsy [2]. Subsequent advances incorporated generative adversarial networks (GANs) [40], [41] to enhance timbre realism, while more recent work using denoising diffusion probabilistic models (DDPMs) [4], [8] has achieved further gains in fidelity and temporal coherence. In parallel, end-to-end systems such as VISinger 1/2 [6], [9] have been developed to generate waveforms directly from musical scores without relying on explicit intermediate features.

Inspired by the success of discrete token modeling in TTS [15], [16], [18], recent SVS studies have explored token-based representations to improve generalization. TokSing [42] employs a non-autoregressive Transformer conditioned on lyrics and pitch embeddings to predict discrete acoustic tokens. HiddenSinger [3] integrates a diffusion-based decoder guided by discrete pitch and semantic tokens, achieving high-quality synthesis. Make-A-Voice [12] unifies speech and singing synthesis through a shared discrete representation, but uses a relatively small proportion of singing data and does not incorporate prompt-based in-context learning. Consequently, it lacks explicit melody conditioning and provides limited flexibility in zero-shot singing scenarios. Recently, models such as Vevo 1.5/2.0 [34] have proposed addressing melody leakage through melody- or prosody-aware tokenizers that explicitly model musical F0 structures. While these methods mitigate melody leakage by redesigning the tokenizer, our work focuses on improving melody control within a prompt-based SVS framework by introducing explicit melody inputs and regulated prompt-based conditioning, enabling more accurate and controllable zero-shot singing voice synthesis.

B. Discrete Token Based Speech Synthesis

Discrete speech modeling has gained momentum following advances in self-supervised learning (SSL) for speech representation. Models such as HuBERT [43] and Wav2Vec 2.0 [44] learn meaningful latent representations from raw audio, which can be quantized into discrete units for downstream tasks. These discrete units provide compact and controllable representations, enabling applications such as low-bitrate speech coding and voice conversion.

Inspired by the success of large language models (LLMs), recent approaches formulate speech synthesis as autoregressive generation over discrete codec tokens. VALL-E [15] pioneered this direction by conditioning on both text and a short acoustic prompt to synthesize high-fidelity speech. Its extensions—VALL-E X [45], VALL-E 2 [46], and VALL-E R [47]—extend the paradigm to cross-lingual synthesis, streaming generation, and improved alignment. SoCodec [20] further improves efficiency through semantic-ordered multi-stream tokenization and segment-level modeling. Through prompt-based conditioning, these models demonstrate strong zero-shot capability and robust speaker generalization.

To reduce inference latency, non-autoregressive (NAR) decoding frameworks have been developed. SoundStorm [21] employs a bidirectional Transformer with confidence-based masked token modeling, generating audio tokens in parallel while maintaining autoregressive-level quality. Multi-token prediction and speculative decoding [48] further accelerate synthesis by predicting multiple codec tokens per decoding step. MaskGCT [18] adopts a masked generative training strategy inspired by masked language modeling, enabling fast and parallel decoding while supporting in-context learning through prompt-aware input masking.

Building on this foundation, we adapt the MaskGCT framework to singing voice synthesis. Our approach incorporates structured melody conditioning and improved melody–timbre control in prompt-based synthesis to address the challenges of pitch fidelity and prosody leakage in zero-shot settings. This design improves controllability over melodic realization while preserving the inference efficiency and generalization strengths of discrete token-based modeling.

C. Prosody and Melody Control in Speech and Singing Voice Synthesis

Fine-grained prosody control—particularly over fundamental frequency (F0) and phoneme duration—is essential for expressive speech and singing synthesis. Several TTS studies have incorporated explicit prosodic supervision to guide model learning. Prosody-TTS [49] augments an end-to-end architecture with auxiliary predictors for phoneme-level F0 and duration, enabling precise rhythm and pitch control without degrading naturalness. [50] adopt utterance-level prosodic features in a hierarchical non-autoregressive model, providing interpretable style modulation across prosodic dimensions while maintaining synthesis quality.

Recent advances extend prosody control to diffusion-based synthesis. DiffStyleTTS [51] combines a diffusion decoder with classifier-free guidance to model prosodic style at both coarse and phoneme-level scales, supporting flexible pitch–duration manipulation. DrawSpeech [52] enables editing by conditioning on user-drawn pitch–energy sketches, which are refined into high-resolution prosody contours.

In SVS, accurate melody control often requires note-level F0 alignment. [53] combine dual-path pitch encoders with local style tokens to capture expressiveness beyond score constraints. Discrete-token SVS frameworks such as TokSing [42] incorporate explicit melody tokens to enrich synthetic singing, while Prompt-Singer [54] decouples vocal range from melody contour in prompt conditioning to preserve pitch accuracy across timbres.

Despite these advances, few systems address conflicts between external melody guidance and prompt-derived timbre cues in discrete-token SVS. We address this gap by introducing coarse-to-fine contrastive learning with frame-level pitch supervision to reduce melody leakage from prompts

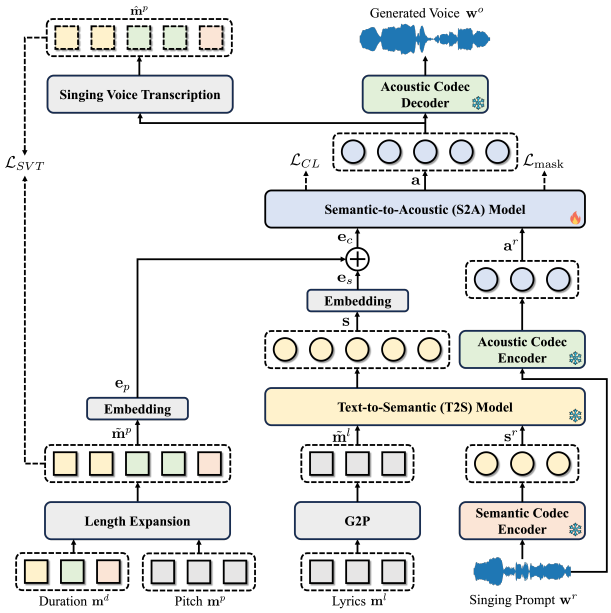


Fig. 3: Overview of CoMelSinger. It adopts a two-stage pipeline: a T2S model encodes lyrics into semantic tokens, and an S2A model generates acoustic tokens conditioned on lyrics, pitch, and prompt. SVT provides pitch supervision. All modules except S2A are frozen during training.

and strengthen external melody control, enabling high-fidelity pitch realization without sacrificing timbre consistency or generalization.

III. Method

A. Overview

To enable zero-shot singing voice synthesis with accurate melody control and disentangled prompt conditioning, we propose *CoMelSinger*, a two-stage framework illustrated in Figure 3. Inspired by MaskGCT [18] and Make-A-Voice [12], CoMelSinger comprises a Text-to-Semantic (T2S) stage and a Semantic-to-Acoustic (S2A) stage. The T2S module f_{T2S} transforms a lyric token sequence $\tilde{\mathbf{m}}^l = [\tilde{m}^l_1, \ldots, \tilde{m}^l_S] \in \mathcal{V}^S_{\text{lyr}}$, obtained from a Grapheme-to-Phoneme (G2P) converter, and a semantic prompt $\mathbf{s}^r = E_S(\mathbf{w}^r)$ extracted from the reference waveform $\mathbf{w}^r$, into a semantic token sequence $\mathbf{s} \in \mathcal{V}^L_{\text{sem}}$, where $\mathcal{V}_{\text{sem}}$ denotes the semantic vocabulary and L the sequence length. The S2A module f_{S2A} then predicts acoustic tokens $\mathbf{a} \in \mathcal{V}^{L\times N}_{\text{aco}}$, conditioned on the semantic tokens $\mathbf{s}$, an acoustic prompt $\mathbf{a}^r = E_A(\mathbf{w}^r) \in \mathcal{V}^{L_r\times N}_{\text{aco}}$, and a regulated pitch sequence $\tilde{\mathbf{m}}^p \in \mathcal{V}^L_{\text{pit}}$. Here, N denotes the number of residual vector quantization (RVQ) codebooks in the acoustic codec, and the regulated pitch sequence $\tilde{\mathbf{m}}^p$ is derived from the pitch $\mathbf{m}^p$ and duration $\mathbf{m}^d$ sequence through the length expansion module $LE(\cdot)$. Both the semantic and acoustic tokens are produced using discrete codec tokenizers following the MaskGCT setup, and the final waveform $\mathbf{w}^o$ is reconstructed from acoustic tokens via the decoder D_A. The complete pipeline is summarized as:

$$\begin{aligned} \mathbf{s}^r &= E_S(\mathbf{w}^r), \quad \mathbf{a}^r = E_A(\mathbf{w}^r), \\ \mathbf{s} &= f_{\text{T2S}}(\mathbf{m}^l, \mathbf{s}^r), \\ \mathbf{a} &= f_{\text{S2A}}(LE(\mathbf{m}^p, \mathbf{m}^d, L), \mathbf{s}, \mathbf{a}^r), \\ \mathbf{w}^o &= D_A(\mathbf{a}). \end{aligned} \tag{1}$$

To synchronize pitch information with frame-level features, we map phonetic durations onto the frame index space. Given a pitch sequence $\mathbf{m}^p$ with corresponding durations $\mathbf{m}^d$, the total duration is $D = \sum_i m^d_i$. For each pitch token m^p_i, its frame span is derived by rounding the cumulative normalized duration:

$$k_{\text{start},i} = \left\lfloor \tfrac{c_{i-1}}{D} L \right\rceil, \quad k_{\text{end},i} = \left\lfloor \tfrac{c_i}{D} L \right\rceil, \quad c_i = \sum_{j=1}^{i} m^d_j, \tag{2}$$

where L denotes the length of the semantic feature sequence. The frame-aligned pitch sequence $\tilde{\mathbf{m}}^p$ is then obtained by repeating each m^p_i for $n_i = k_{\text{end},i} - k_{\text{start},i}$ frames, ensuring an exact length correspondence with the semantic features.

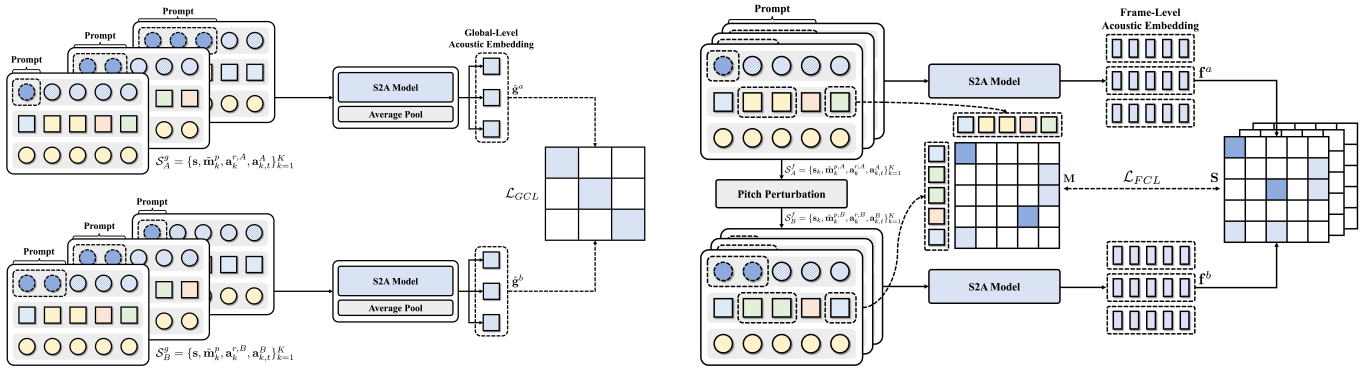
Following MaskGCT [18], CoMelSinger adopts a non-autoregressive masked generative modeling paradigm for both stages. In this framework, the model learns to reconstruct masked tokens within a sequence conditioned on surrounding context and external inputs, rather than generating tokens sequentially. This allows for parallel decoding and better handling of global context compared to traditional autoregressive models. Specifically, we model the conditional probabilities: $p(\mathbf{s} \mid \mathbf{s}_t; \mathbf{m}^l, \mathbf{s}^r; f_{\theta,\text{T2S}})$, $p(\mathbf{a} \mid \mathbf{a}_t; \mathbf{s}, \tilde{\mathbf{m}}^p, \mathbf{a}^r; f_{\theta,\text{S2A}})$. Here, $\mathbf{s}_t$ and $\mathbf{a}_t$ are the partially masked semantic and acoustic token sequences, and the generation is conditioned on lyric inputs $\mathbf{m}^l$, pitch-aligned sequence $\tilde{\mathbf{m}}^p$, and prompt tokens $(\mathbf{s}^r, \mathbf{a}^r)$.

Building upon the original MaskGCT architecture, which employs a LLaMA-style Transformer backbone [55], we extend the S2A model by introducing an additional embedding layer for the regulated pitch sequence $\tilde{\mathbf{m}}^p$. This pitch embedding $\mathbf{e}_p$ is element-wise added to the semantic token embedding $\mathbf{e}_s$ to form the composite conditioning input $\mathbf{e}_c = \mathbf{e}_p + \mathbf{e}_s$, thereby enabling the model to incorporate both linguistic and melodic information during acoustic token generation.

To further enhance melody controllability and suppress interference from prompt-induced timbre cues, we propose a coarse-to-fine contrastive learning framework. At the sequence level, a contrastive loss encourages the predicted acoustic token sequence $\mathbf{a}$ to preserve the overall pitch contour defined by $\tilde{\mathbf{m}}^p$. At the frame level, a token-wise contrastive objective aligns fine-grained acoustic features with localized pitch variations, thereby reinforcing frame-level pitch fidelity. In addition, we introduce an auxiliary singing voice transcription (SVT) model, trained to estimate pitch sequences directly from acoustic tokens. The SVT model provides pseudo pitch labels that serve as external supervision during S2A training. This auxiliary signal further improves alignment between the synthesized melody and the target pitch contour.

B. Coarse-to-Fine Contrastive Learning

Contrastive learning has been increasingly adopted in speech and audio modeling to enforce factor-specific con-



(a) Sequence-level Contrastive Learning. (b) Frame-level Contrastive Learning.

Fig. 4: Overview of the coarse-to-fine contrastive learning strategy. (a) Sequence-level contrastive learning encourages timbre consistency across different melodies. (b) Frame-level contrastive learning uses pitch perturbation to enforce local pitch-awareness and disentangle melody from timbre.

sistency while suppressing undesired variations such as speaker identity or prompt interference. For instance, CLAP-Speech [56] applies multi-scale contrastive learning between textual prosody embeddings and corresponding acoustic realizations, improving prosodic expressivity across varying textual contexts. [57] propose a contrastive loss to enhance the modeling of prosodic focus—including F0, duration, and intensity—by encouraging TTS systems to distinguish emphasized from neutral phonetic segments. [58] use contrastive self-supervision to extract prosody-specific embeddings disentangled from speaker identity, which are useful for style transfer and anonymized generation. [59] further explore contrastive pretraining to align textual context with expressive speech realizations, facilitating zero-shot expressive TTS with better generalization.

Motivated by these findings, our method extends contrastive learning to the discrete-token SVS setting by incorporating hierarchical supervision. At the sequence level, we enforce prompt-invariant acoustic consistency conditioned on identical semantic and pitch tokens, encouraging the model to preserve global melody shape. At the local level, a frame-wise contrastive loss is applied to align acoustic token features with fine-grained pitch variations. This coarse-to-fine scheme allows our model to disentangle pitch conditioning from acoustic prompts and enhances melody fidelity under zero-shot generation.

1) Sequence-Level Contrastive Learning: To promote melody-consistent synthesis under varying acoustic prompts, we introduce a sequence-level symmetric contrastive loss. Given a batch of K training samples that share the same semantic token sequence $\mathbf{s}$ but differ in pitch sequences $\tilde{\mathbf{m}}_k^p$, we construct two sets of inputs: $\mathcal{S}_A^g = \{\mathbf{s}, \tilde{\mathbf{m}}_k^p, \mathbf{a}_k^{r,A}, \mathbf{a}_{k,t}^A\}_{k=1}^K$ and $\mathcal{S}_B^g = \{\mathbf{s}, \tilde{\mathbf{m}}_k^p, \mathbf{a}_k^{r,B}, \mathbf{a}_{k,t}^B\}_{k=1}^K$, where $\mathbf{a}_{k,t}^A$ and $\mathbf{a}_{k,t}^B$ denote masked acoustic tokens, $\mathbf{a}_k^{r,A}$ and $\mathbf{a}_k^{r,B}$ denote acoustic prompts sampled from distinct utterances of the same singer, ensuring consistent timbre across the pair. The prompt length is randomly selected from the range $[\min(\lfloor L/4 \rfloor, 5), \lfloor L/2 \rfloor)$, where L is the length of the semantic sequence. Each input is processed by the S2A model to produce acoustic token embeddings $\mathbf{g}^a, \mathbf{g}^b \in \mathbb{R}^{K \times L \times D}$, which are mean-pooled across the time dimension to yield global acoustic representations $\tilde{\mathbf{g}}^a, \tilde{\mathbf{g}}^b \in \mathbb{R}^{K \times D}$.

To align acoustic outputs with the shared pitch condition while remaining invariant to prompt variation, we apply a symmetric contrastive loss (SCE) [60] defined as:

$$\mathcal{L}_{\text{SCL}} = \frac{1}{2K} \sum_{i=1}^{K} \left(\log \frac{\exp\left(\tilde{\mathbf{g}}_i^a \cdot \tilde{\mathbf{g}}_i^b / \tau\right)}{\sum_{j=1}^{K} \exp\left(\tilde{\mathbf{g}}_i^a \cdot \tilde{\mathbf{g}}_j^b / \tau\right)} + \log \frac{\exp\left(\tilde{\mathbf{g}}_i^b \cdot \tilde{\mathbf{g}}_i^a / \tau\right)}{\sum_{j=1}^{K} \exp\left(\tilde{\mathbf{g}}_i^b \cdot \tilde{\mathbf{g}}_j^a / \tau\right)} \right), \tag{3}$$

where τ is a temperature hyperparameter. Each positive pair $(\tilde{\mathbf{g}}_i^a, \tilde{\mathbf{g}}_i^b)$ corresponds to index-aligned embeddings generated under identical semantic tokens $\mathbf{s}$ and regulated pitch sequence $\tilde{\mathbf{m}}^p$, but conditioned on different acoustic prompts $\mathbf{a}_i^{r,A}$ and $\mathbf{a}_i^{r,B}$. These prompts are randomly selected from non-overlapping segments of the same singer's recordings, ensuring consistent timbre while introducing natural variation. In contrast, off-diagonal pairs $(\tilde{\mathbf{g}}_i^a, \tilde{\mathbf{g}}_j^b)$ for $i \neq j$ serve as negatives due to mismatched pitch sequences, even though semantic tokens and speaker identity remain the same. These negatives are nontrivial, as they reflect realistic melodic differences under otherwise comparable contextual and timbral conditions. This design encourages the model to focus on capturing global pitch structure while remaining invariant to prompt-induced variability.

While this sequence-level objective enforces high-level melodic consistency, it does not explicitly supervise token-level alignment. We therefore complement it with a frame-level contrastive loss to further enhance fine-grained melody control.

2) Frame-Level Contrastive Learning: While the sequence contrastive loss promotes utterance-level melody consistency, it does not directly enforce fine-grained pitch alignment at the frame level—an essential factor in singing synthesis due to rapid and expressive melodic changes. To address this limitation, we introduce a frame-level contrastive objective that supervises token-wise alignment between generated acoustic representations and the input pitch contour. Our design is

motivated by recent work such as CTAP [61], which leverage contrastive learning to align discrete phoneme sequences with speech features for TTS, voice conversion, and ASR tasks under limited supervision. Although our formulation differs in both granularity and modality—operating on pitch tokens rather than phonemes, and targeting melody alignment in singing synthesis—these studies underscore the effectiveness of contrastive supervision for bridging symbolic and acoustic representations. By extending this idea to the SVS domain, our frame-level contrastive loss enhances local pitch fidelity while remaining robust to prompt-induced variation, thereby enabling more precise and expressive melody control in zero-shot scenarios.

Given a batch of K training samples with distinct semantic token sequences $\mathbf{s}_k$ and corresponding pitch sequences $\mathbf{m}_k^p$, we construct two sets of inputs: $\mathcal{S}_A^f = \{\mathbf{s}_k, \tilde{\mathbf{m}}_k^{p,A}, \mathbf{a}_k^{r,A}, \mathbf{a}_{k,t}^A\}_{k=1}^K$ and $\mathcal{S}_B^f = \{\mathbf{s}_k, \tilde{\mathbf{m}}_k^{p,B}, \mathbf{a}_k^{r,B}, \mathbf{a}_{k,t}^B\}_{k=1}^K$, where $\mathbf{a}_{k,t}^A$ and $\mathbf{a}_{k,t}^B$ denote the masked acoustic tokens, $\tilde{\mathbf{m}}_k^{p,A}$ and $\tilde{\mathbf{m}}_k^{p,B}$ denote the regulated pitch sequences derived from the original pitch tokens $\mathbf{m}_k^p$ and their perturbed variants $P(\mathbf{m}_k^p)$, respectively. The acoustic prompts $\mathbf{a}_k^{r,A}$ and $\mathbf{a}_k^{r,B}$ are sampled from different utterances of the same singer to ensure consistent timbre across pairs. The perturbation function $P(\cdot)$ offsets 50% of pitch tokens by integers randomly sampled from $[-6, 6]$, while preserving the original duration sequence $\mathbf{m}^d$.

Each input is passed through the S2A model to produce frame-level acoustic embeddings $\mathbf{f}^a, \mathbf{f}^b \in \mathbb{R}^{K\times L\times D}$. For each training sample k, we compute a cosine similarity matrix $\mathbf{S}^k \in \mathbb{R}^{L\times L}$ between $\mathbf{f}^a$ and $\mathbf{f}^b$. To supervise the similarity learning, we define a soft label matrix $\mathbf{Y}^k \in [-1, 1]^{L\times L}$ capturing pitch and semantic alignment:

$$Y_{ij}^k = \begin{cases} 1, & \text{if } \tilde{m}_{k,i}^{p,A} = \tilde{m}_{k,j}^{p,B} \wedge s_i = s_j \\ \alpha, & \text{if } \tilde{m}_{k,i}^{p,A} = \tilde{m}_{k,j}^{p,B} \wedge s_i \neq s_j \\ 0, & \text{if } \tilde{m}_{k,i}^{p,A} \neq \tilde{m}_{k,j}^{p,B} \\ -1, & \text{if either frame is silence or padding} \end{cases}, \quad (4)$$

where $\alpha \in (0, 1)$ is a tunable coefficient that softly down-weights semantically mismatched but pitch-aligned pairs. The frame-level contrastive loss is formulated as a masked regression objective:

$$\mathcal{L}_{\text{FCL}} = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{L}\sum_{j=1}^{L} \mathbb{1}\left[Y_{ij}^k \geq 0\right] \cdot \left(S_{ij}^k - Y_{ij}^k\right)^2, \quad (5)$$

where $\mathbb{1}[\cdot]$ is an indicator function that masks out invalid entries (e.g., silence or padding).

This formulation encourages the model to produce highly similar acoustic embeddings when both pitch and semantic content align, moderately similar embeddings when only pitch aligns, and dissimilar embeddings otherwise. The similarity is computed within each utterance because the vocal range is typically locally bounded, making repeated pitch tokens more likely. In contrast, pitch overlap across utterances is rare and thus excluded. Additionally, even within the same utterance, repeated pitches may be associated with different semantic tokens, resulting in subtle acoustic variation. Our soft labeling mechanism accounts for this by assigning intermediate similarity, thereby avoiding over-penalization while promoting melody-consistent synthesis. Hence, the final contrastive learning objective is as follows:

$$\mathcal{L}_{\text{CL}} = \lambda_{\text{SCL}} \cdot \mathcal{L}_{\text{SCL}} + \lambda_{\text{FCL}} \cdot \mathcal{L}_{\text{FCL}}, \quad (6)$$

where λ_{SCL} and λ_{FCL} are weighting coefficients that balance the contributions of sequence-level and frame-level supervision, respectively.

C. Singing Voice Transcription for Pitch Guidance

Recent studies have explored the use of Singing Voice Transcription (SVT) to support singing voice synthesis (SVS). For example, ROSVOT [62] proposes a robust SVT model to produce high-quality pitch annotations for large-scale singing datasets, thereby improving SVS performance by enhancing training data quality. However, such approaches treat SVT as an independent preprocessing tool, disconnected from the synthesis process.

In contrast, we integrate the SVT module directly into the training pipeline to provide explicit frame-level pitch supervision, thereby enhancing melody modeling and alignment. Specifically, the SVT model predicts frame-wise discrete pitch tokens from acoustic codec representations, which are then compared against the ground-truth pitch sequence. This supervision enforces alignment between the generated acoustic tokens and the intended melody, encouraging consistent pitch realization, particularly under zero-shot conditions. Furthermore, as the SVT module operates entirely on discrete representations, it is naturally compatible with our codec-based SVS framework and does not require raw audio or continuous F0 contours.

The SVT model adopts a lightweight encoder-only Transformer architecture. Given a sequence of acoustic tokens $\mathbf{a}$, it predicts the corresponding pitch token sequence $\hat{\mathbf{m}}^p$ of length L. The encoder consists of four Transformer layers with a hidden size of 512 and eight attention heads. Each input frame comprises 12 discrete acoustic codes, which are individually embedded, concatenated, and projected to a 512-dimensional representation, followed by layer normalization. The resulting embedding sequence is then passed through a linear classification head to predict a pitch token for each frame. The model is trained using a standard cross-entropy loss between the predicted and reference pitch sequences.

To provide frame-level supervision for pitch modeling, we leverage the pretrained SVT model as a pitch predictor to generate pseudo labels in the form of pitch token sequences $\tilde{\mathbf{m}}^p$, which are temporally aligned with the acoustic frames. As the primary training objective, we apply a cross-entropy loss $\mathcal{L}_{\text{CE}}$ between the predicted pitch tokens $\hat{\mathbf{m}}^p$ and the SVT-derived ground-truth $\tilde{\mathbf{m}}^p$, encouraging accurate token-level classification. However, the cross-entropy objective alone does not account for the temporal continuity inherent in repeated pitch tokens. This often results in jittery predictions, fragmented note segments, and rhythmically unstable outputs.

1) Segment Transition Loss: Prior studies have highlighted the importance of modeling temporal structure for natural singing synthesis. XiaoiceSing [38] introduces syllable-level

duration modeling to preserve rhythmic consistency, while Singing-Tacotron [63] enhances segmental alignment through transition tokens and duration-informed attention mechanisms.

Motivated by these findings, we propose a segment transition loss $\mathcal{L}_{\text{seg}}$ to impose structural regularity on the predicted pitch token sequence. Let $\mathbf{p} \in \mathbb{R}^{L\times C}$ denote the predicted frame-level pitch token distribution obtained by applying a softmax to the decoder logits, where L is the number of frames and C is the size of the pitch token vocabulary. The loss is defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{seg}} = \sum_{t=2}^{L} \Big[(1-b_t)\cdot \|\mathbf{p}_t - \mathbf{p}_{t-1}\|^2 \\ + b_t \cdot \max\left(0, \delta - \|\mathbf{p}_t - \mathbf{p}_{t-1}\|\right)^2,\end{aligned} \tag{7}$$

where $b_t = \mathbb{1}[\tilde{m}_t^p \neq \tilde{m}_{t-1}^p]$ is a binary indicator marking ground-truth pitch boundaries, and δ is a fixed margin that enforces dissimilarity across transitions. This formulation penalizes minimal variation within sustained pitch regions while promoting sharper contrast at pitch change boundaries, thereby enhancing segment continuity and expressive phrasing in the synthesized output.

2) Soft Duration Loss: Inspired by recent advances in speech synthesis that emphasize the importance of temporal alignment and duration modeling [38], [64], we introduce a soft duration loss $\mathcal{L}_{\text{dur}}$ to enhance the rhythmic fidelity of frame-level pitch predictions. Prior works such as FastSpeech [64] and XiaoiceSing [38] employ explicit duration predictors or auxiliary alignment modules to supervise temporal structures. While effective, these methods often introduce architectural overhead or struggle to generalize in expressive singing scenarios. In contrast, our approach provides a fully differentiable supervision signal by directly supervising the temporal distribution of pitch token probabilities using the softmax outputs of the model, without requiring any external duration modeling.

Given a symbolic duration sequence $\mathbf{m}^d = [m_1^d, \ldots, m_S^d]$, we normalize it into a frame-level allocation $\mathbf{a}^d = [a_1^d, \ldots, a_S^d]$, where each element is computed as $a_i^d = \left\lfloor \frac{m_i^d}{D} \cdot L \right\rfloor$, where $D = \sum_{i=1}^{S} m_i^d$ and L denotes the total number of frames. Let $\mathbf{p} \in \mathbb{R}^{L\times C}$ denote the predicted frame-level pitch token distribution obtained via softmax, where C is the size of the pitch vocabulary. For each target pitch token m_i^p, we define its soft duration as the cumulative probability mass $\mathbf{p}_t[m_i^p]$ over its allocated segment of length a_i^d. The soft duration loss is given by

$$\mathcal{L}_{\text{dur}} = \sum_{i=1}^{S} \left(\sum_{t=T_i}^{T_i + a_i^d - 1} \mathbf{p}_t[m_i^p] - a_i^d \right)^2, \tag{8}$$

where $T_i = \sum_{j=1}^{i-1} a_j^d$ denotes the starting frame index for the ith pitch token.

This formulation encourages the model to allocate appropriate probability mass to each pitch token across time, thereby promoting temporally coherent and rhythmically faithful melody generation. The final training objective for the SVT module combines the cross-entropy loss, segment transition loss, and soft duration loss:

$$\mathcal{L}_{\text{SVT}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{seg}} \cdot \mathcal{L}_{\text{seg}} + \lambda_{\text{dur}} \cdot \mathcal{L}_{\text{dur}}. \tag{9}$$

Algorithm 1 Finetuning S2A with Contrastive Learning and SVT Supervision

Require: S2A model parameters θ; frozen SVT model f_{SVT}; training set $\mathcal{D}$; loss weights $\lambda_{\text{CL}}, \lambda_{\text{SVT}}, \lambda_{\text{mask}}$; number of epochs N; learning rate η
Ensure: Trained parameters θ

for $i = 1$ to N **do**
  Sample batch $\mathcal{B} = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ from $\mathcal{D}$
  Split $\mathcal{B}$ into $\mathcal{S}_A^g = \{\mathbf{x}_1, \ldots, \mathbf{x}_{K_g}\}$ and $\mathcal{S}_A^f = \{\mathbf{x}_{K_g+1}, \ldots, \mathbf{x}_K\}$
  Construct $\mathcal{S}_B^g = \mathcal{S}_A^g$, $\mathcal{S}_B^f \leftarrow P(\mathcal{S}_A^f)$
  Define $\mathcal{B}' \leftarrow \mathcal{S}_B^g \cup \mathcal{S}_B^f$
  $\mathbf{a}^r \leftarrow \texttt{PromptGen}(\mathcal{B})$, $\mathbf{a}^{r'} \leftarrow \texttt{PromptGen}(\mathcal{B}')$
  $\tilde{\mathcal{B}} \leftarrow \mathcal{B} \cup \mathbf{a}^r$, $\tilde{\mathcal{B}}' \leftarrow \mathcal{B}' \cup \mathbf{a}^{r'}$
  Forward pass:
    $\mathbf{e} \leftarrow \hat{f}_{\text{S2A}}(\tilde{\mathcal{B}})$, $\mathbf{e}' \leftarrow \hat{f}_{\text{S2A}}(\tilde{\mathcal{B}}')$
    Split $\mathbf{e}$ into $\mathbf{g}^a = \mathbf{e}_{1:K_g}$ and $\mathbf{f}^a = \mathbf{e}_{K_g+1:K}$
    Split $\mathbf{e}'$ into $\mathbf{g}^b = \mathbf{e}'_{1:K_g}$ and $\mathbf{f}^b = \mathbf{e}'_{K_g+1:K}$
    $\hat{\mathbf{g}}^a \leftarrow \texttt{AvgPool}(\mathbf{g}^a)$, $\hat{\mathbf{g}}^b \leftarrow \texttt{AvgPool}(\mathbf{g}^b)$
  Contrastive losses:
    Compute $\mathcal{L}_{\text{SCL}}$ from $\hat{\mathbf{g}}^a$ and $\hat{\mathbf{g}}^b$ according to (3)
    Compute $\mathcal{L}_{\text{FCL}}$ from $\mathbf{f}^a$ and $\mathbf{f}^b$ according to (5)
    Compute $\mathcal{L}_{\text{CL}}$ according to (6)
  Mask prediction and SVT loss:
    $\hat{\mathbf{a}} \leftarrow f_{\text{head}}(\mathbf{f}^a)$
    $\mathcal{L}_{\text{mask}} \leftarrow \texttt{MaskLoss}(\hat{\mathbf{a}}, \mathbf{a})$
    $\hat{\mathbf{m}}^p \leftarrow f_{\text{SVT}}(\texttt{StopGrad}(\hat{\mathbf{a}}))$
    Compute $\mathcal{L}_{\text{SVT}}$ from $\hat{\mathbf{m}}^p$ and $\tilde{\mathbf{m}}^p$ according to (9)
  Total loss and update:
    $\mathcal{L} \leftarrow \lambda_{\text{CL}} \cdot \mathcal{L}_{\text{CL}} + \lambda_{\text{SVT}} \cdot \mathcal{L}_{\text{SVT}} + \lambda_{\text{mask}} \cdot \mathcal{L}_{\text{mask}}$
    $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}$
end for

D. Training and Inference Procedures

We begin by training the Singing Voice Transcription (SVT) model to provide frame-level pitch supervision for subsequent Semantic-to-Acoustic (S2A) adaptation. The SVT model is optimized using a cross-entropy loss $\mathcal{L}_{\text{CE}}$ between the regulated pitch token sequence $\tilde{\mathbf{m}}^p$ and the acoustic token sequence $\mathbf{a}$, thereby learning to predict temporally aligned pitch trajectories from acoustic inputs. Once trained, the SVT module is frozen to serve as a fixed auxiliary supervisor during S2A training.

We then fine-tune the S2A model built upon the MaskGCT framework [18], which leverages masked acoustic modeling for non-autoregressive generation. The training objective for the S2A model comprises three components: (1) the mask token prediction loss $\mathcal{L}_{\text{mask}}$, which reconstructs randomly masked acoustic tokens from noisy inputs using a masked denoising objective; (2) a coarse-to-fine contrastive loss $\mathcal{L}_{\text{CL}}$, composed of sequence-level ($\mathcal{L}_{\text{SCL}}$) and frame-level ($\mathcal{L}_{\text{FCL}}$) terms, to enforce consistency between the melody condition and generated acoustic tokens while mitigating prosody

leakage from the prompt; and (3) an auxiliary SVT loss $\mathcal{L}_{\text{SVT}}$, which encourages the predicted acoustic tokens to be rhythmically and melodically consistent with the SVT-inferred pitch contour. The fine-tuning algorithm for the S2A model is summarized in Algorithm 1. Each training sample $\mathbf{x}_k \in \mathcal{B}$ comprises a semantic sequence $\mathbf{s}_k$, a regulated pitch sequence $\tilde{\mathbf{m}}_k^p$, and a time-aligned acoustic token sequence $\mathbf{a}_{k,t}$.

To improve adaptation efficiency and reduce overfitting, we apply Low-Rank Adaptation (LoRA) [65] to fine-tune the S2A model's diffusion estimator module, which is implemented using a DiffLlama-style architecture. LoRA introduces trainable low-rank matrices into the linear layers of pretrained models, enabling efficient fine-tuning by updating only a small subset of parameters while keeping the original weights frozen. This allows our model to retain prior knowledge from the TTS domain while adapting to the stylistic nuances of singing voice synthesis with limited data.

During inference, we follow the parallel iterative decoding introduced in MaskGCT [18] to generate acoustic token sequences. Unlike the original setup, which requires either an explicit duration input or a learned duration predictor to determine the output length, we directly use the ground-truth duration sequence $\mathbf{m}^d$ extracted from the input music score. This allows precise control over the number of generated tokens—equal to the total duration $D = \sum_i m_i^d$—thus preserving the intended temporal structure of the synthesized performance.

IV. Experimental Setups

A. Dataset

We conduct experiments on two publicly available mandarin singing corpora: M4Singer [23][2] and Opencpop [22][3]. The M4Singer dataset comprises studio-quality recordings from 20 professional singers spanning SATB vocal ranges, along with comprehensive annotations including lyrics, pitch, note duration, and slur information. The Opencpop dataset contains 100 Chinese pop songs sung by a professional female vocalist, with precise phoneme, note, and syllable-level annotations aligned to the music score. For evaluation, we construct both seen- and unseen-singer test sets. For seen-singer evaluation, we randomly select 50 utterances each from the M4Singer and Opencpop datasets. For zero-shot (unseen-singer) evaluation, we use 10 male and 10 female singers from the OpenSinger [66][4] dataset. Since OpenSinger lacks complete music score annotations, we pair it with M4Singer's score sequences to enable evaluation. All audio is uniformly down-sampled to 24 kHz with 16-bit quantization.

To fine-tune the S2A model, we preprocess the dataset to obtain temporally aligned semantic tokens $\mathbf{s}$, acoustic tokens $\mathbf{a}$, and regulated pitch tokens $\tilde{\mathbf{m}}^p$. All audio segments are first converted to mono and resampled to 24 kHz. We then utilize the pretrained semantic and acoustic codec models from the MaskGCT framework[5] to extract $\mathbf{s}$ and $\mathbf{a}$. For samples with mismatched token lengths, we apply zero-padding to align their temporal dimensions for frame-level supervision. The regulated pitch sequence $\tilde{\mathbf{m}}^p$ is derived by expanding the original pitch sequence $\mathbf{m}^p$ according to the duration sequence $\mathbf{m}^d$, as described in Section III-A. The SVT model is trained using the preprocessed acoustic tokens $\mathbf{a}$ and the corresponding regulated pitch tokens $\tilde{\mathbf{m}}^p$.

B. Implementation Details

The SVT model is trained on a single NVIDIA RTX A5000 GPU using the AdamW optimizer with a learning rate of 1e−5, weight decay of 0.01, and a cosine learning rate schedule with 5K warm-up steps over 50K updates. Training is performed for 100 epochs with a batch size of 32 using mixed-precision (FP16) computation. Subsequently, the S2A model is fine-tuned on four NVIDIA RTX A5000 GPUs with data parallelism. We adopt the AdamW optimizer with a learning rate of 1e−5, 32K warm-up steps, and the inverse square root learning rate schedule. Fine-tuning is conducted for 300K steps with a total batch size of 32, where the first 8 samples are used for sequence contrastive learning and the remaining 24 for frame-level contrastive learning. We apply dropout (0.1), label smoothing (0.1), and gradient clipping to stabilize training. During this stage, all model components are frozen except the S2A decoder. The loss weights are set as follows: $\lambda_{\text{SCL}} = 0.5$, $\lambda_{\text{FCL}} = 1.0$, $\lambda_{\text{CL}} = 0.1$, $\lambda_{\text{seg}} = 0.5$, $\lambda_{\text{dur}} = 0.3$, and $\lambda_{\text{SVT}} = 0.2$.

C. Evaluation Metrics

1) Objective Evaluation: To quantify the performance of our system in terms of pitch accuracy, timbre consistency, and perceptual quality, we conduct objective evaluations under both seen-singer and zero-shot settings. The following metrics are employed:

a) Mel-Cepstral Distortion (MCD): MCD is used to evaluate spectral fidelity by computing the frame-wise Euclidean distance between mel-cepstral coefficients of the synthesized and reference audio. It serves as a proxy for spectral similarity, where lower values indicate more accurate spectral reconstruction and reduced distortion.

b) Fundamental Frequency RMSE (F0-RMSE): F0-RMSE measures pitch prediction accuracy by calculating the root mean squared error between the fundamental frequency (F0) trajectories of the generated and reference waveforms. A lower F0-RMSE reflects better alignment with the intended melody and more precise pitch control.

c) Speaker Embedding Cosine Similarity (SECS): To assess timbre similarity, we compute cosine similarity between speaker embeddings extracted from the synthesized and reference audio using a WavLM-based speaker verification model [67][6]. SECS values range from 0 to 1, with higher scores indicating closer alignment in vocal identity.

[2] https://github.com/M4Singer/M4Singer

[3] https://xinshengwang.github.io/opencpop/

[4] https://github.com/Multi-Singer/Multi-Singer.github.io?tab=readme-ov-file

[5] https://github.com/open-mmlab/Amphion/tree/main/models/tts/maskgct

[6] https://huggingface.co/microsoft/wavlm-base-sv

TABLE I: The prosody similarity between synthesized and prompt speech in terms of differences in pitch, energy, and other prosodic indicators. Lower values indicate higher similarity.

LibriTTS	**Pitch**				**Energy**				**Others**		
	Mean ↓	Std ↓	Skew ↓	Kurt ↓	Mean ↓	Std ↓	Skew ↓	Kurt ↓	Jitter ↓	Shimmer ↓	HNR ↓
Paired	**19.03**	**22.79**	**2.43**	**19.68**	**1.80**	**1.54**	**0.33**	**0.95**	**0.63**	**0.56**	**3.28**
Unpaired	53.41	32.47	2.81	21.82	4.60	2.28	0.46	1.23	0.89	0.78	3.79
AISHELL-3	**Pitch**				**Energy**				**Others**		
	Mean ↓	Std ↓	Skew ↓	Kurt ↓	Mean ↓	Std ↓	Skew ↓	Kurt ↓	Jitter ↓	Shimmer ↓	HNR ↓
Paired	**42.70**	**26.29**	**1.13**	**3.85**	**2.94**	**2.38**	**0.44**	**1.71**	**0.78**	**0.43**	**4.42**
Unpaired	65.73	29.87	1.63	7.05	5.05	2.58	0.57	1.83	0.94	0.58	4.95

TABLE II: Evaluation results on the seen test set for singing voice synthesis. Subjective metrics are reported with 95% confidence intervals. GT stands for Ground Truth.

Model	**Subjective Evaluations**			**Objective Evaluations**			
	MOS-Q ↑	MOS-N ↑	SMOS ↑	MCD ↓	F0-RMSE ↓	SingMOS ↑	SECS ↑
GT	4.17 ± 0.16	4.38 ± 0.18	4.41 ± 0.14	-	-	4.37	0.925
GT (Acoustic Codec)	4.01 ± 0.22	4.19 ± 0.19	4.48 ± 0.12	0.93	0.012	4.31	0.906
DiffSinger [4]	3.68 ± 0.20	3.79 ± 0.15	3.86 ± 0.16	4.59	0.084	4.13	0.769
VISinger2 [9]	3.59 ± 0.22	3.86 ± 0.16	3.91 ± 0.16	5.36	0.061	4.15	0.792
StyleSinger [7]	3.67 ± 0.15	3.92 ± 0.21	4.11 ± 0.16	4.95	0.112	4.19	0.833
SPSinger [13]	3.81 ± 0.18	**4.10 ± 0.12**	4.06 ± 0.17	4.28	0.054	4.28	0.860
Vevo 1.5 [34]	3.85 ± 0.12	3.96 ± 0.16	4.17 ± 0.16	4.18	0.051	**4.39**	0.907
CoMelSinger (ours)	**3.90 ± 0.16**	4.02 ± 0.12	**4.22 ± 0.15**	**4.17**	**0.042**	4.32	**0.912**

d) SingMOS: SingMOS (Singing Mean Opinion Score) [68][7] is a learned metric trained to predict human perceptual ratings of singing voice quality. It is based on a curated dataset of professional listening tests, in which human raters assign mean opinion scores to both natural and synthesized singing in Chinese and Japanese, addressing the scarcity of large-scale perceptual annotations in the singing domain. SingMOS produces scores in the range of 0 to 5, with higher values indicating greater perceived naturalness and overall quality. As a reference-free metric, it enables scalable automatic evaluation in zero-shot and low-resource conditions.

2) Subjective Evaluation: To assess perceptual quality, we conducted a Mean Opinion Score (MOS) evaluation with 20 participants who have formal training in singing and experience in vocal performance[8]. Each participant rated the synthesized samples based on overall naturalness (MOS-N), audio quality (MOS-Q), and timbre similarity (SMOS). A 5-point Likert scale was used, where a score of 5 indicates excellent perceptual quality and 1 denotes poor quality.

V. Experimental Results

A. Evaluating Prompt-Induced Prosody Similarity in MaskGCT

Several recent TTS systems [19], [69], [70] have reported high prosodic similarity between the speech prompt and the synthesized output. While this may appear beneficial in TTS, it reveals a form of prosody leakage, where expressive cues from the prompt inadvertently influence the generated speech. This issue becomes particularly problematic in singing voice synthesis (SVS), where pitch and rhythm should be governed solely by the input music score. As discussed in Section I, we refer to this phenomenon as prosody leakage.

To investigate whether MaskGCT [18] exhibits such behavior, we conduct a prosody similarity analysis following prior evaluation protocols. NaturalSpeech 2 and 3 [19] quantify prosodic similarity by comparing pitch and duration features between the prompt and output, while StyleTTS-ZS [70] computes Pearson correlation coefficients of acoustic features to evaluate prosodic alignment.

Inspired by these approaches, we evaluate the prosodic similarity of MaskGCT in both English and Mandarin using the LibriTTS [25][9] and AISHELL [71][10] datasets, respectively. For each speaker, we randomly sample 50 utterances to construct the test sets. During inference, we synthesize 50 utterances per dataset by conditioning on the same target text but using different speech prompts. We then compute the acoustic-level similarity between each synthesized utterance and: (1) its paired prompt (i.e., the one used during generation), and (2) an unpaired prompt from the same speaker. This comparison allows us to quantify the extent of prompt-induced prosody similarity, which serves as an indicator of potential prosody leakage in the model.

Table I presents a quantitative analysis of prompt-induced prosody similarity by comparing the acoustic differences between synthesized and prompt speech under paired and

[7] https://github.com/South-Twilight/SingMOS

[8] This study has been approved by the Department Ethics Review Committee (DERC) at the National University of Singapore under DERC Ref Code: 000479.

[9] https://www.openslr.org/60/

[10] https://openslr.org/93/

TABLE III: Subjective (with 95% confidence intervals) and objective evaluation results on the unseen test set for zero-shot SVS. Note that MCD is excluded since ground-truth alignments are unavailable in this setting.

Model	Subjective Evaluations			Objective Evaluations		
	MOS-Q ↑	MOS-N ↑	SMOS ↑	F0-RMSE ↓	SingMOS ↑	SECS ↑
GT	4.20 ± 0.12	4.35 ± 0.14	4.55 ± 0.15	-	4.41	0.932
GT (Acoustic Codec)	4.07 ± 0.18	4.22 ± 0.15	4.32 ± 0.11	0.015	4.66	0.921
DiffSinger	3.75 ± 0.16	3.72 ± 0.18	3.25 ± 0.12	0.098	4.11	0.658
VISinger2	3.72 ± 0.19	3.74 ± 0.20	3.31 ± 0.16	0.074	4.08	0.704
StyleSinger	3.48 ± 0.11	3.82 ± 0.18	3.85 ± 0.15	0.125	4.22	0.853
SPSinger	**3.92 ± 0.15**	4.03 ± 0.15	3.76 ± 0.10	0.065	**4.29**	0.844
Vevo 1.5 [34]	3.72 ± 0.14	3.81 ± 0.12	4.02 ± 0.15	0.094	4.16	0.870
CoMelSinger (ours)	3.87 ± 0.18	**4.11 ± 0.15**	**4.14 ± 0.14**	**0.048**	4.25	**0.897**

TABLE IV: Objective evaluation results on the seen test set for singing voice synthesis, comparing CoMelSinger with representative systems that employ supervised pitch information for melody control. GT denotes Ground Truth.

Model	MCD ↓	F0-RMSE ↓	SingMOS ↑
CoMelSinger (ours)	4.17	0.042	**4.32**
XiaoiceSing	4.54	0.052	4.26
SingAug	**4.16**	**0.035**	3.96
RMSSinger	4.33	0.077	4.15

unpaired conditions. Across both LibriTTS and AISHELL-3, paired prompts consistently yield lower differences in pitch, energy, and other prosodic indicators, confirming stronger alignment in prosodic patterns. In contrast, the unpaired condition results in noticeably higher deviations, particularly in pitch mean, energy mean, and jitter, suggesting that the synthesized outputs are heavily influenced by the prosodic characteristics of the prompt.

B. Seen Singer Singing Voice Synthesis

For the seen singer evaluation, we compare CoMelSinger with five strong baseline systems: DiffSinger [4], VISinger2 [9], SPSinger [13], StyleSinger [7], and Vevo 1.5 [34]. To ensure a fair comparison, all models adopt HiFi-GAN [72] as the vocoder during both training and inference. As presented in Table II, CoMelSinger achieves the highest scores across both subjective and objective metrics, demonstrating its capability to synthesize natural and expressive singing voices from seen singers.

We first note that the performance gap between the Ground Truth (GT) and GT with Acoustic Codec is minimal across all metrics, confirming that the discrete acoustic token representation introduces negligible degradation and establishing a strong upper bound for token-based SVS systems. CoMelSinger approaches this bound closely, suggesting that its improvements arise from architectural designs—particularly the disentangled modeling of melody and timbre—rather than signal-level enhancements. Among all models, CoMelSinger achieves the highest SMOS and SECS scores, indicating strong timbre consistency and accurate preservation of speaker identity, which validates the effectiveness of the in-context prompting mechanism. It also attains the lowest F0-RMSE and one of the highest SingMOS scores, reflecting precise melody reproduction and high perceptual naturalness. Furthermore, its competitive MCD score demonstrates the model's ability to reconstruct spectral features with smooth and consistent vocal quality, confirming the effectiveness of the proposed structured melody control strategy.

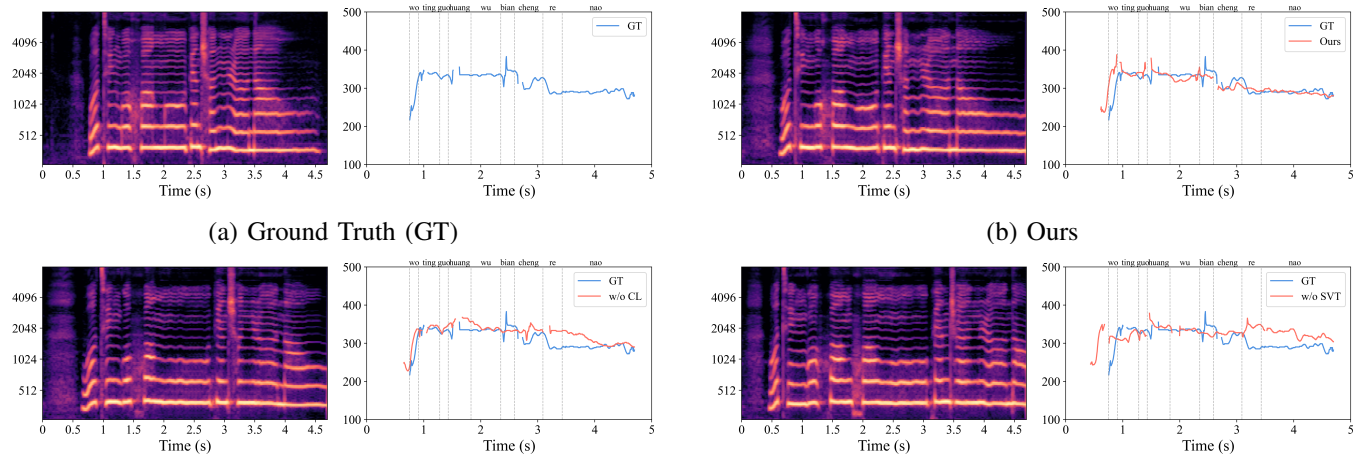
In addition, we compare CoMelSinger with several representative SVS systems that employ supervised pitch-aware conditioning under the seen-singer setting, including XiaoiceSing [38], SingAug [73], and RMSSinger [74]. XiaoiceSing enables precise melody control through explicit F0 modeling with residual log-F0 prediction, SingAug enhances pitch modeling in SVS by applying pitch-based and mix-up data augmentation during training, and RMSSinger achieves melody control by modeling pitch directly from realistic music scores using a diffusion-based pitch modeling approach. As shown in Table IV, CoMelSinger achieves competitive objective performance against these supervised systems, with comparable MCD and F0-RMSE and the highest SingMOS score. Although SingAug attains slightly lower MCD and F0-RMSE, these baseline systems rely on singer-dependent training and are not designed for zero-shot SVS, whereas CoMelSinger maintains strong performance while supporting zero-shot generalization.

C. Zero-Shot Singing Voice Synthesis

We further evaluate CoMelSinger in a zero-shot setting, where the model synthesizes singing voices from speakers not seen during training. As shown in Table III, CoMelSinger maintains strong performance across all subjective and objective metrics, exhibiting only minimal degradation compared to the seen condition.

In contrast, baseline systems show notable declines in key timbre- and melody-related metrics such as SMOS, SECS, and F0-RMSE, underscoring their limited ability to generalize to unseen vocal identities. CoMelSinger's robustness in zero-shot scenarios is attributed to the synergy between in-context prompting—which leverages short acoustic references to anchor timbre—and large-scale speech pretraining, which imparts transferable prosodic priors.

Despite the inherent challenge of handling unseen timbres, CoMelSinger continues to achieve high speaker similarity while preserving accurate pitch trajectories. This balance between identity retention and melodic fidelity demonstrates



(a) Ground Truth (GT) (b) Ours

(c) w/o Contrastive Loss (CL) (d) w/o SVT

Fig. 5: Visualization of mel-spectrograms and pitch contours for the ground truth, the proposed model, and ablated variants. The predicted pitch trajectory (red) is overlaid with the ground-truth pitch (blue), with word-level boundaries indicated by vertical dashed lines and pinyin annotations.

TABLE V: Objective evaluation results for the ablation study. CL represents the coarse-to-fine contrastive learning strategy, where SCL and FCL respectively represents sequence and frame-level contrastive learning, SVT represents using SVT for pitch guidance. The "-w/o CL+SVT" configuration corresponds to the MaskGCT-based SVS baseline.

Model	MCD ↓	F0-RMSE ↓	SingMOS ↑	SECS ↑
CoMelSinger	**4.17**	**0.042**	**4.32**	**0.912**
-w/o CL	4.91	0.080	4.12	0.895
-w/o SCL	4.53	0.062	4.25	0.900
-w/o FCL	4.82	0.075	4.18	0.892
-w/o SVT	5.53	0.194	3.95	0.883
-w/o CL + SVT	5.89	0.210	3.83	0.874

the model's strong generalization capacity. While many existing approaches face trade-offs between controllability and naturalness, CoMelSinger effectively reconciles both through its structured architecture and explicit conditioning scheme. These findings position CoMelSinger as a strong baseline for zero-shot singing voice synthesis with discrete representations.

D. Ablation Study

a) Component Analysis: To assess the contribution of each component in CoMelSinger, we perform ablation studies by systematically disabling key modules. Table V reports results on four objective metrics: MCD, F0-RMSE, SingMOS, and SECS. In particular, the configuration "w/o CL + SVT" corresponds to the MaskGCT-based SVS baseline, where the S2A module is fine-tuned without explicit melody control or prosody disentanglement. Removing the entire coarse-to-fine contrastive learning (CL) framework leads to substantial degradation across all metrics, indicating reduced pitch accuracy and speaker consistency. This highlights the importance of contrastive objectives in disentangling pitch from timbre and improving input–output alignment.

To isolate the effects of each contrastive branch, we further ablate sequence contrastive learning (SCL) and frame-level contrastive learning (FCL) individually. Excluding SCL moderately affects MCD and SECS, suggesting its role in maintaining global speaker identity. In contrast, removing FCL causes a larger drop in F0-RMSE and SingMOS, confirming its effectiveness in modeling fine-grained pitch details and promoting melodic continuity. These results validate the hierarchical design of our contrastive learning framework.

We also evaluate the impact of the singing voice transcription (SVT) module, which provides auxiliary pitch supervision. Excluding SVT results in higher F0-RMSE and lower SingMOS, confirming the benefit of explicit alignment signals for structured melody control. The most severe degradation occurs when both CL and SVT are removed, indicating their complementary roles in pitch–timbre disentanglement and temporal stability.

Figure 5 visualizes mel-spectrograms and pitch contours for the ground truth, our model, and two ablated variants. Predicted pitch trajectories (red) are overlaid with ground-truth pitch (blue), with word-level boundaries marked by dashed lines and pinyin annotations. Compared to the ablated models, CoMelSinger achieves better pitch alignment and smoother contours, illustrating the effectiveness of both CL and SVT in preserving melodic structure.

b) SVT Evaluation and Analysis: To assess the reliability of the SVT module, we conduct an explicit frame-level evaluation using token accuracy, precision, recall, and F1 score across multiple training configurations, as summarized in Table VI. We evaluate SVT models trained on M4Singer, Opencpop, their combination, and MIR-ST500 [75]. MIR-ST500 is a large-scale singing transcription dataset comprising over 160k annotated notes from 500 pop songs. Training on the combined M4Singer and Opencpop datasets achieves the best overall performance, indicating robust pitch token prediction across diverse singing styles.

TABLE VI: Frame-level evaluation results of the SVT model under different training configurations. Accuracy, precision, recall, and F1 score are reported.

Data Configuration	Accuracy	Precision	Recall	F1
M4Singer + Opencpop	**0.790**	**0.719**	**0.707**	**0.711**
M4Singer	0.749	0.691	0.674	0.680
Opencpop	0.707	0.506	0.485	0.489
MIR-ST500	0.725	0.449	0.390	0.405
MIR-ST500 pretrain + Combined finetune	0.648	0.575	0.542	0.548

TABLE VII: Performance of various fine-tuning strategies with differing trainable parameter ratios.

Method	MCD ↓	F0-RMSE ↓	SingMOS ↑	SECS ↑	Trainable (%)
FT-LoRA	4.26	**0.053**	**4.34**	**0.920**	6.51%
FT-LLRD	4.33	0.069	4.31	0.894	100.00%
FT-Pitch	4.47	0.062	5.95	0.902	4.46%
FT-Prefix	4.52	0.059	6.13	0.911	5.25%
FT-PGS	**4.21**	0.084	4.26	0.887	16.63%∼16.63%
FT-Full	4.41	0.099	4.13	0.859	100.00%

Models trained on individual datasets exhibit degraded performance, particularly on Opencpop and MIR-ST500, which can be attributed to domain mismatch and limited coverage of singing pitch patterns. We further investigate cross-dataset generalization by pretraining SVT on MIR-ST500 followed by fine-tuning on the combined dataset. Although this strategy improves precision and recall compared to training on MIR-ST500 alone, its overall performance remains inferior to training directly on singing-specific data, underscoring the importance of domain-relevant supervision for accurate pitch modeling.

c) Comparison of Fine-Tuning Strategies: We evaluate six representative fine-tuning strategies on the DiffLlama backbone, each trained for 1000 epochs under identical schedules. The comparison highlights trade-offs between parameter efficiency, adaptation capacity, and overfitting risk under limited SVS data.

- **FT-LoRA**: Applies LoRA to self-attention projections with $r = 16$, $\alpha = 32$, and dropout 0.1. Only the pitch encoder, output head, and cond module are trainable.
- **FT-LLRD**: Freezes DiffLlama and fine-tunes pitch/output/cond modules with layer-wise learning rates decayed from bottom to top: $\eta_\ell = \eta_0 \cdot \gamma^{L-1-\ell}$.
- **FT-Pitch**: Fine-tunes only the pitch encoder, output head, and cond module; the backbone remains frozen.
- **FT-Prefix**: Adds 20 virtual tokens to each DiffLlama layer using prefix tuning (shared across 16 layers, injected into attention and MLP). Pitch/output/cond modules are also fine-tuned.
- **FT-PGS**: Unfreezes two upper DiffLlama layers every 200 epochs, progressively increasing trainable capacity.
- **FT-Full**: Fully fine-tunes all model parameters, including the entire DiffLlama backbone.

Table VII presents a comparison of six fine-tuning strategies in terms of both objective and subjective performance, along with their respective trainable parameter ratios. FT-LoRA delivers the best overall performance, achieving the lowest F0-RMSE, highest SingMOS, and highest SECS, while updating only 4.81% of the parameters—highlighting the effectiveness of low-rank adaptation for efficient model tuning. FT-PGS achieves the lowest MCD, suggesting enhanced spectral fidelity through gradual unfreezing, though its pitch accuracy is affected by delayed optimization of lower layers. FT-Prefix and FT-Pitch yield consistent results with minimal overhead, demonstrating the utility of lightweight adaptation modules. In contrast, FT-LLRD and FT-Full fine-tune all parameters yet underperform across most metrics, indicating that full-capacity adaptation may lead to overfitting or instability in data-scarce settings. These results underscore that parameter-efficient strategies, particularly LoRA, can match or surpass full-model fine-tuning while substantially reducing computational cost.

VI. Conclusion

In this work, we present CoMelSinger, a zero-shot singing voice synthesis framework that extends discrete token-based TTS models to support structured and controllable melody generation. Built upon the non-autoregressive MaskGCT architecture, CoMelSinger incorporates lyrics and pitch tokens as inputs, enabling fine-grained alignment between the musical score and the generated voice. To address the challenge of prosody leakage from prompt-based conditioning—an issue unique to singing synthesis—we propose a coarse-to-fine contrastive learning strategy that explicitly disentangles pitch information from the timbre prompt. Furthermore, we introduce a lightweight singing voice transcription (SVT) module to provide frame-level pitch and duration supervision, enhancing the model's ability to follow the intended melody with precision. Extensive experiments on both seen and unseen singers demonstrate that CoMelSinger achieves strong zero-shot generalization, consistently outperforming competitive SVS baselines in pitch accuracy, timbre consistency, and subjective quality. Our results confirm that structured melody control and contrastive disentanglement are essential for scalable and expressive singing synthesis. We believe CoMelSinger opens new possibilities for discrete token-based SVS, enabling scalable and zero-shot singing generation.

References

[1] W. Guo, Y. Zhang, C. Pan, R. Huang, L. Tang, R. Li, Z. Hong, Y. Wang, and Z. Zhao, "Techsinger: Technique controllable multilingual singing voice synthesis via flow matching," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 22, 2025, pp. 23 978–23 986.

[2] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Sinsy: A deep neural network-based singing voice synthesis system," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2803–2815, 2021.

[3] J.-S. Hwang, S.-H. Lee, and S.-W. Lee, "Hiddensinger: High-quality singing voice synthesis via neural audio codec and latent diffusion models," *Neural Networks*, vol. 181, p. 106762, 2025.

[4] J. Liu, C. Li, Y. Ren, F. Chen, and Z. Zhao, "Diffsinger: Singing voice synthesis via shallow diffusion mechanism," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 10, 2022, pp. 11 020–11 028.

[5] Z. Ye, W. Xue, X. Tan, J. Chen, Q. Liu, and Y. Guo, "Comospeech: One-step speech and singing voice synthesis via consistency model," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 1831–1839.

[6] Y. Zhang, J. Cong, H. Xue, L. Xie, P. Zhu, and M. Bi, "Visinger: Variational inference with adversarial learning for end-to-end singing voice synthesis," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7237–7241.

[7] Y. Zhang, R. Huang, R. Li, J. He, Y. Xia, F. Chen, X. Duan, B. Huai, and Z. Zhao, "Stylesinger: Style transfer for out-of-domain singing voice synthesis," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 597–19 605.

[8] J. Zhao, L. Q. H. Chetwin, and Y. Wang, "Sintechsvs: A singing technique controllable singing voice synthesis system," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2641–2653, 2024.

[9] Y. Zhang, H. Xue, H. Li, L. Xie, T. Guo, R. Zhang, and C. Gong, "Visinger2: High-fidelity end-to-end singing voice synthesis enhanced by digital signal processing synthesizer," in *Interspeech 2023*, 2023, pp. 4444–4448.

[10] D.-M. Byun, S.-B. Kim, and S.-W. Lee, "Hierarchical diffusion model for zero-shot singing voice synthesis with midi priors," *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

[11] D.-M. Byun, S.-H. Lee, J.-S. Hwang, and S.-W. Lee, "Midi-voice: Expressive zero-shot singing voice synthesis via midi-driven priors," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 12 622–12 626.

[12] R. Huang, C. Zhang, Y. Wang, D. Yang, L. Liu, Z. Ye, Z. Jiang, C. Weng, Z. Zhao, and D. Yu, "Make-a-voice: Unified voice synthesis with discrete representation," *arXiv preprint arXiv:2305.19269*, 2023.

[13] J. Zhao, C. Low, and Y. Wang, "Spsinger: Multi-singer singing voice synthesis with short reference prompt," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[14] S. Dai, Y. Wang, R. B. Dannenberg, and Z. Jin, "Everyone-can-sing: Zero-shot singing voice synthesis and conversion with speech reference," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[15] S. Chen, C. Wang, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Neural codec language models are zero-shot text to speech synthesizers," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 33, pp. 705–718, 2025.

[16] Z. Du, Q. Chen, S. Zhang, K. Hu, H. Lu, Y. Yang, H. Hu, S. Zheng, Y. Gu, Z. Ma *et al.*, "Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens," *arXiv preprint arXiv:2407.05407*, 2024.

[17] Z. Du, Y. Wang, Q. Chen, X. Shi, X. Lv, T. Zhao, Z. Gao, Y. Yang, C. Gao, H. Wang *et al.*, "Cosyvoice 2: Scalable streaming speech synthesis with large language models," *arXiv preprint arXiv:2412.10117*, 2024.

[18] Y. Wang, H. Zhan, L. Liu, R. Zeng, H. Guo, J. Zheng, Q. Zhang, X. Zhang, S. Zhang, and Z. Wu, "Maskgct: Zero-shot text-to-speech with masked generative codec transformer," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[19] Z. Ju, Y. Wang, K. Shen, X. Tan, D. Xin, D. Yang, E. Liu, Y. Leng, K. Song, S. Tang, Z. Wu, T. Qin, X. Li, W. Ye, S. Zhang, J. Bian, L. He, J. Li, and S. Zhao, "Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models," in *Forty-first International Conference on Machine Learning, ICML 2024*. OpenReview.net, 2024.

[20] H. Guo, F. Xie, K. Xie, D. Yang, D. Guo, X. Wu, and H. Meng, "Socodec: A semantic-ordered multi-stream speech codec for efficient language model based text-to-speech synthesis," in *2024 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2024, pp. 645–651.

[21] Z. Borsos, M. Sharifi, D. Vincent, E. Kharitonov, N. Zeghidour, and M. Tagliasacchi, "Soundstorm: Efficient parallel audio generation," *arXiv preprint arXiv:2305.09636*, 2023.

[22] Y. Wang, X. Wang, P. Zhu, J. Wu, H. Li, H. Xue, Y. Zhang, L. Xie, and M. Bi, "Opencpop: A high-quality open source chinese popular song corpus for singing voice synthesis," in *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022*. ISCA, 2022, pp. 4242–4246.

[23] L. Zhang, R. Li, S. Wang, L. Deng, J. Liu, Y. Ren, J. He, R. Huang, J. Zhu, X. Chen *et al.*, "M4singer: A multi-style, multi-singer and musical score provided mandarin singing corpus," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6914–6926, 2022.

[24] S. Dai, Y. Wu, S. Chen, R. Huang, and R. B. Dannenberg, "Singstyle111: A multilingual singing dataset with style transfer," in *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023*, 2023, pp. 765–773.

[25] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A corpus derived from librispeech for text-to-speech," in *20th Annual Conference of the International Speech Communication Association, Interspeech 2019*. ISCA, 2019, pp. 1526–1530.

[26] H. He, Z. Shang, C. Wang, X. Li, Y. Gu, H. Hua, L. Liu, C. Yang, J. Li, P. Shi *et al.*, "Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation," in *IEEE Spoken Language Technology Workshop, SLT 2024*. IEEE, 2024, pp. 885–890.

[27] J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen *et al.*, "Libri-light: A benchmark for asr with limited or no supervision," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020*. IEEE, 2020, pp. 7669–7673.

[28] Z. Wu, Q. Li, S. Liu, and Q. Yang, "Dctts: Discrete diffusion model with contrastive learning for text-to-speech generation," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 11 336–11 340.

[29] H. Tang, X. Zhang, J. Wang, N. Cheng, and J. Xiao, "Avqvc: One-shot voice conversion by vector quantization with applying contrastive learning," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4613–4617.

[30] K. Qian, Y. Zhang, S. Chang, M. Hasegawa-Johnson, and D. Cox, "Unsupervised speech decomposition via triple information bottleneck," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7836–7846.

[31] X. Zhao, F. Liu, C. Song, Z. Wu, S. Kang, D. Tuo, and H. Meng, "Disentangling content and fine-grained prosody information via hybrid asr bottleneck features for voice conversion," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7022–7026.

[32] J. Lian, C. Zhang, and D. Yu, "Robust disentangled variational speech representation learning for zero-shot voice conversion," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6572–6576.

[33] J. Zhao, X. Wang, and Y. Wang, "Prosody-adaptable audio codecs for zero-shot voice conversion via in-context learning," *arXiv preprint arXiv:2505.15402*, 2025.

[34] X. Zhang, J. Zhang, Y. Wang, C. Wang, Y. Chen, D. Jia, Z. Chen, and Z. Wu, "Vevo2: Bridging controllable speech and singing voice generation via unified prosody learning," *arXiv preprint arXiv:2508.16332*, 2025.

[35] H. Kenmochi and H. Ohshita, "Vocaloid-commercial singing synthesizer based on sample concatenation," in *Interspeech*, vol. 2007, 2007, pp. 4009–4010.

[36] J. Bonada and X. Serra, "Synthesis of the singing voice by performance sampling and spectral models," *IEEE signal processing magazine*, vol. 24, no. 2, pp. 67–79, 2007.

[37] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "An HMM-based singing voice synthesis system," in *Proc. Interspeech 2006*, 2006, pp. paper 2077–Thu1BuP.7.

[38] P. Lu, J. Wu, J. Luan, X. Tan, and L. Zhou, "Xiaoicesing: A high-quality and integrated singing voice synthesis system," in *21st Annual Conference of the International Speech Communication Association, Interspeech 2020*. ISCA, 2020, pp. 1306–1310.

[39] Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.-Y. Liu, "Deepsinger: Singing voice synthesis with data mined from the web," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1979–1989.

[40] P. Chandna, M. Blaauw, J. Bonada, and E. Gómez, "Wgansing: A multi-voice singing voice synthesizer based on the wasserstein-gan," in *2019 27th European signal processing conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

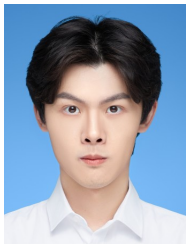
[41] R. Huang, C. Cui, F. Chen, Y. Ren, J. Liu, Z. Zhao, B. Huai, and Z. Wang, "Singgan: Generative adversarial network for high-fidelity singing voice generation," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 2525–2535.

[42] Y. Wu, C. Zhang, J. Shi, Y. Tang, S. Yang, and Q. Jin, "Toksing: Singing voice synthesis based on discrete tokens," in *25th Annual Conference of the International Speech Communication Association, Interspeech 2024*. ISCA, 2024.

[43] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 29, pp. 3451–3460, 2021.

[44] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations,"

Advances in neural information processing systems, vol. 33, pp. 12 449–12 460, 2020.
[45] Z. Zhang, L. Zhou, C. Wang, S. Chen, Y. Wu, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Speak foreign languages with your own voice: Cross-lingual neural codec language modeling," *arXiv preprint arXiv:2303.03926*, 2023.
[46] S. Chen, S. Liu, L. Zhou, Y. Liu, X. Tan, J. Li, S. Zhao, Y. Qian, and F. Wei, "Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers," *arXiv preprint arXiv:2406.05370*, 2024.
[47] B. Han, L. Zhou, S. Liu, S. Chen, L. Meng, Y. Qian, Y. Liu, S. Zhao, J. Li, and F. Wei, "Vall-e r: Robust and efficient zero-shot text-to-speech synthesis via monotonic alignment," *arXiv preprint arXiv:2406.07855*, 2024.
[48] T. D. Nguyen, J.-H. Kim, J. Choi, S. Choi, J. Park, Y. Lee, and J. S. Chung, "Accelerating codec-based speech synthesis with multi-token prediction and speculative decoding," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.
[49] G. Pamisetty and K. Sri Rama Murty, "Prosody-tts: An end-to-end speech synthesis system with prosody control," *Circuits, Systems, and Signal Processing*, vol. 42, no. 1, pp. 361–384, 2023.
[50] T. Raitio, J. Li, and S. Seshadri, "Hierarchical prosody modeling and control in non-autoregressive parallel neural tts," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7587–7591.
[51] J. Liu, Z. Liu, Y. Hu, Y. Gao, S. Zhang, and Z. Ling, "Diffstyletts: Diffusion-based hierarchical prosody modeling for text-to-speech with diverse and controllable styles," in *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025*. Association for Computational Linguistics, 2025, pp. 5265–5272.
[52] W. Chen, S. Yang, G. Li, and X. Wu, "Drawspeech: Expressive speech synthesis using prosodic sketches as control conditions," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.
[53] J. Lee, H.-S. Choi, and K. Lee, "Expressive singing synthesis using local style token and dual-path pitch encoder," *arXiv preprint arXiv:2204.03249*, 2022.
[54] Y. Wang, R. Hu, R. Huang, Z. Hong, R. Li, W. Liu, F. You, T. Jin, and Z. Zhao, "Prompt-singer: Controllable singing-voice-synthesis with natural language prompt," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024*. Association for Computational Linguistics, 2024, pp. 4780–4794.
[55] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.
[56] Z. Ye, R. Huang, Y. Ren, Z. Jiang, J. Liu, J. He, X. Yin, and Z. Zhao, "CLAPSpeech: Learning prosody from text context with contrastive language-audio pre-training," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Jul. 2023, pp. 9317–9331.
[57] S. Latif, I. Kim, I. Calapodescu, and L. Besacier, "Controlling prosody in end-to-end TTS: A case study on contrastive focus generation," in *Proceedings of the 25th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Nov. 2021, pp. 544–551.
[58] J. Weston, R. Lenain, U. Meepegama, and E. Fristed, "Learning de-identified representations of prosody from raw audio," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 134–11 145.
[59] Y. Xiao, X. Wang, X. Tan, L. He, X. Zhu, S. Zhao, and T. Lee, "Contrastive context-speech pretraining for expressive text-to-speech synthesis," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 2099–2107.
[60] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 322–330.
[61] C. Qiang, H. Li, Y. Tian, R. Fu, T. Wang, L. Wang, and J. Dang, "Learning speech representation from contrastive token-acoustic pretraining," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 10 196–10 200.
[62] R. Li, Y. Zhang, Y. Wang, Z. Hong, R. Huang, and Z. Zhao, "Robust singing voice transcription serves synthesis," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*. Association for Computational Linguistics, 2024, pp. 9751–9766.
[63] T. Wang, R. Fu, J. Yi, Z. Wen, and J. Tao, "Singing-tacotron: Global duration control attention and dynamic filter for end-to-end singing voice synthesis," in *Proceedings of the 1st International Workshop on Deepfake Detection for Audio Multimedia*, 2022, pp. 53–59.
[64] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," *Advances in neural information processing systems*, vol. 32, 2019.
[65] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.
[66] R. Huang, F. Chen, Y. Ren, J. Liu, C. Cui, and Z. Zhao, "Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3945–3954.
[67] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
[68] Y. Tang, J. Shi, Y. Wu, and Q. Jin, "Singmos: An extensive open-source singing voice dataset for mos prediction," *arXiv preprint arXiv:2406.10911*, 2024.
[69] K. Shen, Z. Ju, X. Tan, E. Liu, Y. Leng, L. He, T. Qin, S. Zhao, and J. Bian, "Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers," in *The Twelfth International Conference on Learning Representations, ICLR 2024*. OpenReview.net, 2024.
[70] Y. A. Li, X. Jiang, C. Han, and N. Mesgarani, "StyleTTS-ZS: Efficient high-quality zero-shot text-to-speech synthesis with distilled time-varying style diffusion," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Association for Computational Linguistics, Apr. 2025, pp. 4725–4744.
[71] Y. Shi, H. Bu, X. Xu, S. Zhang, and M. Li, "Aishell-3: A multi-speaker mandarin tts corpus and the baselines," *arXiv preprint arXiv:2010.11567*, 2020.
[72] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in neural information processing systems*, vol. 33, pp. 17 022–17 033, 2020.
[73] S. Guo, J. Shi, T. Qian, S. Watanabe, and Q. Jin, "Singaug: Data augmentation for singing voice synthesis with cycle-consistent training strategy," in *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022, Incheon, Korea, September 18-22, 2022*. ISCA, 2022, pp. 4272–4276.
[74] J. He, J. Liu, Z. Ye, R. Huang, C. Cui, H. Liu, and Z. Zhao, "Rmssinger: Realistic-music-score based singing voice synthesis," in *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*. Association for Computational Linguistics, 2023, pp. 236–248.
[75] J. Wang and J. R. Jang, "On the preparation and validation of a large-scale dataset of singing transcription," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*. IEEE, 2021, pp. 276–280.



Junchuan Zhao Junchuan Zhao is currently a Ph.D. student at the School of Computing, National University of Singapore, advised by Prof. Ye Wang. He received his B.Sc. Degree in Telecommunications Engineering with Management from Beijing University of Posts and Telecommunications in 2022 and M.Sc. degree in Computer Science from National University of Singapore in 2023. His research centers on generative modeling for speech and multimodal content, with a particular focus on speech and singing voice synthesis, neural speech codecs, and talking head generation.

Wei Zeng Wei Zeng is currently pursuing the Ph.D. degree at the Sound and Music Computing Lab, National University of Singapore, under the supervision of Prof. Ye Wang. He received dual bachelor's degrees in Energy and Power Engineering and Musicology from Shanghai Jiao Tong University, and the master's degree from the National University of Singapore. His research interests lie in music information retrieval and machine learning, with particular emphasis on automatic piano transcription, singing voice transcription, and expressive performance rendering.

Tianle Lyu Tianle Lyu received the B.Eng. degree in Computer Science from Chongqing University of Posts and Telecommunications in 2025. He is currently a research intern at the Sound of Music Computing Lab, National University of Singapore, under the supervision of Ye Wang. His research interests include computer vision, machine learning, and large language models.

Ye Wang is an Associate Professor in the Computer Science Department at the National University of Singapore (NUS). He received his Ph.D. degree from Tampere University of Technology in Finland in 2002, M.Sc. degree from Braunschweig University of Technology in Germany in 1993, and B.Sc. degree from South China University of Technology in China in 1983. He established and directed the sound and music computing (SMC) Lab. Before joining NUS, he was a member of the technical staff at Nokia Research Center in Tampere, Finland for 9 years. His research philosophy is that technology should be developed for good - such as expanding access, increasing affordability, and improving quality of healthcare and education. Guided by this philosophy, he explored a new programmatic research agenda, which became his signature research in the past decade: cognitive neuroscience-inspired Sound and Music Computing for Human Health and Potential (SMC4HHP), attempting to address two big questions. 1) How to enable users to discover their preferred music that satisfies clinical requirements for Rhythmic Auditory Stimulation (RAS) based gait rehabilitation and exercise via music search, recommendation and generation? 2) How to leverage on the relationship between speech and singing to build applications for speech intervention? To address the above questions, he led the development of MusicRx technologies to make RAS accessible and affordable, and of SLIONS for speech intervention for various populations.